λ-Game System

λ-Game System

舛本 現
東京大学総合文化研究科広域システム科学系
gen@sacral.c.u-tokyo.ac.jp

概 要

ゲームにおける「ルール」と「戦略」の関係を考えたい．そのためにゲームの枠組みをλ計算によって表現する．「ルール」と「戦略」といった異なる階層にあると考えられているものを，λ式という同じ枠組で表わすことにより「ルール」と「戦略」の間にある operator と operand という関係が崩れ，元々ルールが想定していた以外のものも戦略として使えるようになる．それらの戦略はルールに干渉し，ルールが元々用意していなかった利得を得ることが可能になる．このような状況をモデル化し，その時のゲームの様子を議論する

1 はじめに

1.1 戦略の open-endedness とルールの openness

ゲームとプレイの違いを考えたい．ゲームには決ったルールがあり，各プレイヤーはルールが規定する戦略のなかでどの戦略が自分の利得を最大にするかを考え，選択し，それに応じた利得を得る．一方，ここで考えたいプレイとは，子供の遊びのようなものであり，そこではルールは一応決まっているはずなのだが，ルールでは想定されてない新しい戦略が次々と生み出され，それに対してルールを強引に解釈することによってそれでもなんとかプレイは続いていくのである．

たとえば，子供はゲームのような行動をとることは少なく，ほとんどすべての状況においてここでプレイと呼ぶような interaction をしているのでないだろうか．例えばじゃんけん．これはゲーム論的には 3×3 の利得行列ですべてが表わされるものであるが，子供のじゃんけんにおいては，グーチョキパーを同時に表現してるから無敵だと称する奇妙な手の握りや，繰り返しのじゃんけんでは後出し何回で一回分の負けなどといったローカルルールがどんどん生み出される．

人工生命の研究においてはゲームにおける戦略の open-end な進化は重要なテーマであり，これまでも数多くの研究がなされてきた [10]．しかしルールが rigid に決まり，そのなかでの最適な行動が存在するのなら open-end な時間発展など不可能であり，あとに残るのは，本当は最適な戦略があるのだが人間の合理性や計算能力が限られているからそこに到達しえない，という問題だけになってしまう．そこで多くの open-end な時間発展を扱う研究では，戦略の空間に openness を (非常に限定されたかたちで) 導入した．それはある時は記憶戦略における記憶の長さ ([7, 11]) であり，また各個体のつくる相手のモデルの決まらなさ ([8, 9, 13]) であったりした．

しかし他方にはルールにこそ戦略を open-end に発展させる源があるという見方がある．これはつまりルールが open であることこそが戦略の open-endedness を生み出すという立場である．例えば，社会におけるルールというのは多くの場合，自然言語で記述されており，自然言語であればもちろん解釈は開かれているから，現実の社会における戦略は多様でありえると考える．ルールの openness を導入したモデルでは，例えば Howard の meta-game theory のように囚人のジレン

マにおいて「あなたが協調するなら私も協調し，あなたが裏切るなら私も裏切る」といったルールに言及するメタ戦略を (one-shot でも) 許すような枠組を用いたり [5, 6]，あるいは Hofstadter が紹介した Nomic というゲーム ([4]) のようにすべてのルールを明示的に自然言語で書き，それを更新していくゲームを考えたりすることが行われる．

1.2 関数と変数

ではルールの openness というのはどのようにして可能になるのであろうか．ルールは戦略の定義域を規定してそれらの組に対する利得を決定し，各プレイヤーはルールの定義域の範囲内の戦略で利得の最適化を図る．このような点において，ここで考えているゲームにおけるルールと戦略というのは，関数と変数のような関係にあるといえよう．変数 (戦略) は決して関数 (ルール) の規定する定義域の外に出ることはできないし，関数 (ルール) も定義域の外の入力 (戦略) に対しては出力 (得点) を与えることはできない．

一方，プレイでは定義域が開かれているので，ルールは定義域の外の入力に対しても出力を返さなくてはいけない．ここで要求されているのは，定義域を限定せずにどんな入力に対してもとりあえずは出力を返す体系である．そのためには定義域のクラスが関数自体のクラスと同じくらい広く，関数と変数，すなわちルールと戦略，が同じ format で書かれている type-free なものでなければいけない．

ここではゲームを記述するために λ 計算を導入する．λ 計算は type-free な計算体系であり，このことはすなわち同じ λ 式が関数と変数のどちらにもなれることを意味している．また λ 計算ではゲームの利得表を記述するのに必要な真偽値や if 文，自然数を表わすことができる．

ここで論じたのと同じように λ 式の関数/変数の duality に注目してそれを化学反応における分子の duality に適用した有名な研究に Fontana と Buss の Alchemy がある [3, 12]．

2 Model

2.1 真偽値，if 文，自然数

λ 計算では，真偽値，if 文，自然数を以下のように表現することができる [2]．(もちろんこれらは λ 計算による if 文や自然数の表現のうちのひとつであり，これら以外の表現も可能である．)

- Boolean values

$$\mathbf{T} \equiv \lambda xy.x \qquad \mathbf{F} \equiv \lambda xy.y \tag{1}$$

- Conditional

$$\text{“if } x \text{ then } P \text{ else } Q\text{”} \equiv \lambda x.xPQ \tag{2}$$

$$\begin{aligned}(\lambda x.xPQ)\,\mathbf{T} \to \mathbf{T}\,P\,Q \to (\lambda xy.x)\,P\,Q \to (\lambda y.P)Q \quad &\to \quad P\\ (\lambda x.xPQ)\,\mathbf{F} \to \mathbf{F}\,P\,Q \to (\lambda xy.y)\,P\,Q \to (\lambda y.y)Q \quad &\to \quad Q\end{aligned}$$

- Numerals(Barendregt)

$$\mathbf{0} \equiv \lambda x.x \;\; (= \mathbf{I}), \tag{3}$$

$$\mathbf{n+1} \equiv \lambda x\,.\,x\mathbf{F}\mathbf{n}, \tag{4}$$

$$\begin{aligned}
\mathbf{succ} &\equiv \lambda xy.y(\lambda xy.y)x \quad (= \lambda xy.y\mathbf{F}x), & (5)\\
\mathbf{pred} &\equiv \lambda x.x(\lambda xy.y) \quad (= \lambda x.x\mathbf{F}), & (6)\\
\mathbf{iszero} &\equiv \lambda x.x(\lambda xy.x) \quad (= \lambda x.x\mathbf{T}). & (7)
\end{aligned}$$

2.2 λ-game system

Fig.1(a) のようなゲームの利得表を持つ one-shot の囚人のジレンマゲームを考える [1]. ここで C, D はそれぞれ協調と裏切りを表わしている. ここでゲームのルールに要求されていることは, 各プレイヤーの戦略の組に対して, それらを区別してそれぞれの場合に応じて利得表に従った数を返すことである. すなわちゲームのルールは Fig.1(b) のように if 文を 2 回使うことによって各プレイヤーの戦略を区別し, 場合分けをして, それぞれの利得を決定している.

そこで前節で導入した真偽値を使って, C(協調) を $\mathbf{T}(= \lambda xy.x)$ で, D(裏切り) を $\mathbf{F}(= \lambda xy.y)$ で表現するとすることにする. すると利得を返す game master の λ 式 $\mathbf{G}$ は, 利得の自然数を Barendregt 数で表わし, 前節の if 文を二重に使うことによって以下の式 (8) に書ける (λ 式のなかで太字で書いてある数字は実際には Barendregt 数の λ 式が入る. それを具体的に書いたのが式 (9) である).

$$\begin{aligned}
\mathbf{G} &\equiv \lambda x.x(\lambda y.y\,\mathbf{3}\,\mathbf{5})(\lambda y.y\,\mathbf{0}\,\mathbf{1}) & (8)\\
&\equiv \lambda x.x(\lambda y.y(\lambda z.z\mathbf{F}(\lambda z.z\mathbf{F}(\lambda z.z\mathbf{FI})))\\
&\qquad\qquad (\lambda z.z\mathbf{F}(\lambda z.z\mathbf{F}(\lambda z.z\mathbf{F}(\lambda z.z\mathbf{F}(\lambda z.z\mathbf{FI}))))) & (9)\\
&\quad (\lambda y.y\ \mathbf{I}\\
&\qquad\qquad (\lambda z.z\mathbf{FI}))
\end{aligned}$$

2 つの戦略 S_{self}, S_{opp} が与えられたとき, それぞれの戦略のとる利得は, $\mathbf{G}$ にこれら S_{self}, S_{opp} を代入することで得られる.

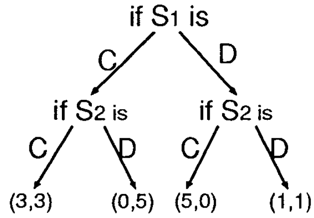
$$\mathbf{G}S_{self}S_{opp} \to \mathbf{G}'_{self}S_{opp} \to Reward_{opp} \qquad (10)$$

$$\mathbf{G}S_{opp}S_{self} \to \mathbf{G}'_{opp}S_{self} \to Reward_{self} \qquad (11)$$

例えば D と C が対戦したときの D の利得は, $\mathbf{GTF} \to (\lambda y.y\mathbf{35})\mathbf{F} \to \mathbf{5}$ と計算される. 他の戦略の組に対しても, $\mathbf{GTT} \to \mathbf{3}$, $\mathbf{GFT} \to \mathbf{0}$, $\mathbf{GFF} \to \mathbf{1}$ となり, この game master($\mathbf{G}$) が Fig.1 の利得表を表わしていることがわかる.

S1 \ S2	C	D
C	3 3	0 5
D	5 0	1 1

(a)



(b)

図 1: Two person one-shot prisoners' dilemma game

3 Numerical Experience

1.2 節で述べたように，λ 計算は type-free な体系なので，前節で定義した game master **G** は実際には **T**, **F** 以外でも任意の戦略 (λ 式) を受け付けることができる．そこで計算機でランダムに λ 式を生成して，それらの λ 式がこのゲームにおいて戦略としてどのような振舞いをするのかを調べた．λ 式の生成の仕方は以下の通りである．

1. Generate new λ-terms(L) by randomly combining atom elements.
   - atomic elements $\equiv \{x_1, \ldots, x_m, \mathbf{T}, \mathbf{F}, \mathbf{I}(= \lambda x.x), \mathbf{S}(= \lambda xyz.xz(yz)), \mathbf{V}(= \lambda xyz.zxy)\}$.
   - λ-term is composed of n times (here: $1 \leq n \leq 20$) application or abstraction of atomic elements recursively.
2. New λ-term L is reduced to a normal form S.
   - λ-terms which do not acquire a normal form within a preset limit (here: 1000) are removed from λ-strategies.

さらにゲームとして成立するための条件として，元のゲームの戦略である **T**, **F** と自分自身に対しては数 (Barendregt の意味で) を返さないといけないという制約を加える．すなわち ($\mathbf{G}S\mathbf{T}$), ($\mathbf{GT}S$), ($\mathbf{G}S\mathbf{F}$), ($\mathbf{GF}S$), ($\mathbf{G}SS$) の 5 つがすべて Barendregt 数にならない λ 式は戦略として認めない．

以上の条件の元で計算機でランダムに λ 式を生成したものから，いくつかの例を Table 1 に示す．**T** と対戦して (4, 4) で協調したり，(6, 0) で搾取したり，さらには **D** と対戦して (2, 1) で勝ったりと元の利得表にない利得のペアを生み出だす多様な戦略が観察された．これらの戦略 λ 式はみな game master の if 文と干渉することによって，言い換えると game master の if 文の中をみて，それを操作することによって，このような得点を得ている．

表 1: Examples of strategies

	examples of λ-terms	vs. **T**	vs. **F**	vs. self	**G**′
1	$\lambda xy.x\ (= \mathbf{T})$	3, 3	0, 5	3	
2	$\lambda xy.y\ (= \mathbf{F})$	5, 0	1, 1	1	
3	$\lambda x_1.x_1(\lambda x_2x_3.x_3\mathbf{F}x_2)\mathbf{F}(\lambda x_4.x_1)$	5, 3	0, 5	5	Type I
4	$\lambda x_1.x_1(\lambda x_2x_3x_4x_5x_6.\mathbf{F}(x_4x_5))\mathbf{SSI}$	5, 0	2, 1	2	
5	$\lambda x_1.(x_1(\lambda x_2.\mathbf{S})\mathbf{VFVF})$	6, 0	2, 1	2	
6	$\lambda x_1x_2.\ x_2\mathbf{TF}$	0, 0	0, 0	0	
7	$\lambda x_1x_2.(x1(\lambda x_3x_4.(x_4(\lambda x_5x_6x_7x_8\ .\ (x_8\mathbf{FS}))x_3))x_2\mathbf{F}))$	2, 0	1, 0	0	Type II
8	$\lambda x_1x_2.(x_1(x_2(\lambda x_3.(x_3(\lambda x_4x_5x_6x_7\ .\ (x_5\mathbf{S})))))))$	0, 2	0, 2	2	
9	$\lambda x_1x_2.(x_2(\lambda x_3x_4.(x_3(\lambda x_5.x_1\mathbf{FS}))))$	0, 4	0, 4	4	
10	$\lambda x_1x_2.(x_1(x_2\mathbf{IF})\mathbf{S}x_2x_1))$	3, 3	0, 0	3	
11	$\lambda x_1x_2.x_2(\lambda x_3.x_3(\lambda x_4x_5x_6x_7x_8\ .\ x_8\mathbf{F}(x_6x_7))\mathbf{T})x_1$	4, 4	0, 6	5	Type III
12	$\lambda x_1x_2.(x_2(\lambda x_3.(x_3(\lambda x_4x_5.(x_1x_5(x_5x_5)(x_4x_5))))))$	4, 2	0, 0	2	

4 Analysis

前節で見られたような，元の利得表にない得点のペアをあたえる戦略は game master の if 文とどのように干渉しているのかを調べる．

4.1 相手の得点を計算する

各戦略 λ 式 S_{self} が相手の得点を計算するときの式 (10)

$$\mathbf{G}S_{self}S_{opp} \to \mathbf{G}'_{self}S_{opp} \to Reward_{opp}$$

において，S_{self} がまず $\mathbf{G}$ と反応してつくる途中の λ 式 $\mathbf{G}'_{self}$ に注目する．この $\mathbf{G}'_{self}$ は自分が相手の得点を計算するときに相手に対してどういう態度をとっているのかを表わしているといえる．$\mathbf{G}'$ には以下の 3 つのタイプがある．Type I は $\mathbf{G}'_{self}$ が if 文のかたちをしているもので，元々のゲームで想定されてる戦略である $\mathbf{T},\mathbf{F}$ がつくるのもこのタイプであり，ランダムにつくった λ 式でゲームとしての制約を満すのものは 8 割以上が Type I になった．Type II は定数関数で，相手の戦略に関係なくある定数を与える．Type I の次に多く生成されランダムに生成された λ 式のうち約 1 割強がこのタイプである．Type III はそれ以外である．

- Type I (conditional form)

  $$\mathbf{G}' = \lambda x.x\,\mathbf{m}\,\mathbf{n} \qquad \mathbf{m},\mathbf{n} : \text{Barendregt numeral}$$

  "if you are x, then I will give you m points, else I'll give you n points"
  Examples : $\lambda x.x\mathbf{01}$, $\lambda x.x\mathbf{35}$, $\lambda x.x\mathbf{15}$, $\lambda x.x\mathbf{21}$, $\cdots$

- Type II (constant function)

  $$\mathbf{G}' \;=\; \lambda x.\mathbf{n} \qquad \mathbf{n} : \text{Barendregt numeral}$$

  " I always give you n points"

  Examples : $\lambda x.\mathbf{0}$, $\lambda x.\mathbf{2}$, $\lambda x.\mathbf{5}$, $\cdots$

- Type III (others)

  Examples : $\lambda x.(x\mathbf{F}(\lambda yz.yz)(x\mathbf{01}))$, $\lambda x.(x\,\mathbf{F}\,\mathbf{2})(x\,\mathbf{0}\,\mathbf{1})$, $\cdots$

4.2 自分の得点を計算する

自分の得点を計算するときには式 (11)

$$\mathbf{G}S_{opp}S_{self} \to \mathbf{G}'_{opp}S_{self} \to Reward_{self}$$

で計算されるが，これは相手のつくる $\mathbf{G}'_{opp}$ に依存する．しかし Type II が相手の場合には自分の λ 式 S_{self} が何であっても，それに関係なく相手が用意した定数が自分の得点になる．そこでここでは Type I の戦略と対戦する場合を考える．

この場合，各戦略 λ 式は 2 つの場面で if 文に代入される．

1. 相手の得点を計算するとき．game master($\mathbf{G}$) に代入されて $\mathbf{G}'_{self}$ をつくる．

2. 自分の得点を計算するとき．相手のつくる $\mathbf{G}'_{opp}$ に代入されて自分の得点を返す.

ここでの if 文とは式 (2) で定義された $\lambda x.xPQ$ (= "if x then P else Q") のかたちをした λ 式である．これらの各局面で，もっとも単純な戦略である $\mathbf{T}$ や $\mathbf{F}$ は，if 文の用意した 2 つの選択肢の内容 (P, Q) をみることはなく，その中身に関係なく片方を選ぶだけである (以下で $*$ は任意の λ 式を表わす).

$$
\begin{aligned}
(\lambda x\,.\,xP*)\mathbf{T} &\rightarrow P \\
(\lambda x\,.\,x*Q)\mathbf{F} &\rightarrow Q
\end{aligned}
$$

それに対して，例えば Table.1 の 4 の戦略 $\lambda x_1.x_1(\lambda x_2x_3x_4x_5x_6.\mathbf{F}(x_4x_5))\mathbf{SSI}$ (= S) は，自分が代入される if 文の内容をみて以下のように振舞いを変えることができる.

$$(\lambda x\,.\,x\,\underline{(\lambda x\,.\,x**)}Q)\,S \rightarrow Q \tag{12}$$

$$(\lambda x\,.\,x\,\underline{\;(\lambda x\,.\,x)\;}\,Q)\,S \rightarrow (\lambda xy.y\mathrm{F}x)Q \quad (= \mathbf{succ}\,Q) \tag{13}$$

つまりこの戦略 λ 式は "if x then P else Q" の P が if 文であるかまたは 1 以上の自然数である場合 ($P = \lambda x.x**$) は Q をそのまま返し (式 (12))，もし P が 0 の場合 ($P = \lambda x.x$) は if 文から **succ**(式 (5) で定義) をつくって Q に作用させることで結果的に $Q+1$ をつくることを意味している (式 (13)).

そして game master(**G**) は 2 重の if 文なので式 (12) の場合に相当し，例えば **F** と対戦して相手のつくる $\mathbf{G}'$ が $\lambda x.x\mathbf{01}$ の時は式 (13) の場合に相当する．よってこの戦略は，相手の得点を計算する時は自分の $\mathbf{G}'_{self}$ として $\lambda x.x\mathbf{01}$ (= "if x then 0 else 1") をつくって **F** のように振舞う一方で自分の得点を計算する時は相手の $\mathbf{G}'_{opp}$ の if 文 ("if x then 0 else 1") の中にある数を操作してそこに書かれているよりも多くの得点を得ることを可能にしたのである.

5 まとめ

各戦略 λ 式は，それぞれ異なった場面で異なった if 文に代入されるというゲームの制約の元で，自分がこれから代入されるそれぞれ if 文の中をみるという形で if 文と干渉している．すなわち自分がこれから代入されるのが game master(**G**) なのか，それとも相手がつくった $\mathbf{G}'_{opp}$ なのかを事前に判断して，それぞれに異なった反応をすることによって game master が用意した元々の利得表にないような利得を獲得している.

さらにいくつかの戦略は game master に代入されるときは，自分の $\mathbf{G}'_{self}$ として単純な **F** のように相手に振舞う if 文を返し，相手がつくった $\mathbf{G}'_{opp}$ が if 文の場合は，その if 文の部分 λ 式から「1 足す (succ)」という関数を合成して，そこに書かれている数字に対して操作をほどこしてから自分の得点にするということによって，高い利得を得ている.

このように λ 計算という type-free な体系でゲームを記述することにより，ルールと戦略という異なった階層にあるものが同じ format で書かれることになり，その結果として，戦略がルールと干渉して cheat することが可能となった.

参考文献

[1] Axelrod, R., *The Evolution of Cooperation*. Basic Books, New York, 1984.

[2] Barendregt, H. G., *The Lambda Calculus: Its Syntax and Semantics*, Amsterdam, North Holland, 1981.

[3] Fontana, W., Buss L. W., The arrival of the fittest: Toward a theory of biological organization, *Bulletin of Mathematical Biology* 56, 1–64, 1994.

[4] Hofstadter, D.R.: Metamagical Themas. BasicBooks (1985).

[5] Howard, N., The theory of meta-games, *General Systems* 11, 167–186, 1966.

[6] Howard, N., *Paradoxes of Rationality: Theory of Metagames and Political Behavior*. MIT Press, Cambridge, MA. 1971.

[7] Ikegami, T., From genetic evolution to emergence of game strategies, *Physica D* 75, 310–327, 1994.

[8] Ikegami, T., Taiji, M., Structures of Possible Worlds in a Game of Players with Internal Models. *Acta Poly. Scan.* MA. 91, 283–292, 1998.

[9] Ikegami, T., Taiji, M., Imitation and Cooperation in Coupled Dynamical Recognizers, in *Advances in Artificial Life.* (eds. Floreano,D. et al.), 545–554, Springer-Verlag, 1999.

[10] Langton, C.G. et al.(eds),*Artificial Life II.* Addison-Wesley, CA. 1992.

[11] Lindgren, K., Evolutionary phenomena in simple dynamics, 295–311, in *Artificial Life II*, Langton, C.G. et al.(eds), Addison-Wesley, CA. 1992.

[12] Pietro Speroni di Fenizio, A Less Abstract Artificial Chemistry, in the proceedings of Artificial Life VII, 49–53, 2000.

[13] Taiji, M. and Ikegami, T., Dynamics of internal models in game players, *Physica D* 134, 253–266. 1999.